

[illegible]

```
FFFFFFFFF 000000 RRRRRRRR FFFFFFFFFF MM MM TTTTTTTTTT CCCCCCCC PPPPPPPP
FFFFFFFFF 000000 RRRRRRRR FFFFFFFFFF MM MM TTTTTTTTTT CCCCCCCC PPPPPPPP
FF 00 00 RR RR FF MM MM TT CC CC PP PP
FF 00 00 RR RR FF MM MM TT CC CC PP PP
FF 00 00 RR RR FF MM MM TT CC CC PP PP
FF 00 00 RR RR FF MM MM TT CC CC PP PP
FFFFFFFFF 00 00 RRRRRRRR FFFFFFFF MM MM TT TT TT TT TT CC CC PPPPPPPP
FFFFFFFFF 00 00 RRRRRRRR FFFFFFFF MM MM TT TT TT TT TT CC CC PPPPPPPP
FF 00 00 RR RR FF MM MM TT TT TT TT TT CC CC PP
FF 00 00 RR RR FF MM MM TT TT TT TT TT CC CC PP
FF 00 00 RR RR FF MM MM TT TT TT TT TT CC CC PP
FF 00 00 RR RR FF MM MM TT TT TT TT TT CC CC PP
FF 000000
FF 000000 RR RR FF MM MM TT CCCCCCCC PP
                                     ....
                                     ....
                                     ....
                                     ....

LL      IIIIII SSSSSSSS
LL      IIIIII SSSSSSSS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SSSSSS
LL      II     SSSSSS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SS
LLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLL IIIIII SSSSSSSS
```



```
1 0001 0 MODULE FOR$$FMTCP (%TITLE'FORTRAN OBJECT TIME FORMAT COMPILER'  
2 0002 0 IDENT = '2-006' ! File: FORFMTCP.B32 Edit: SBL2006  
3 0003 0 ) =  
4 0004 1 BEGIN  
5 0005 1 *****  
6 0006 1 *  
7 0007 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
8 0008 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
9 0009 1 * ALL RIGHTS RESERVED.  
10 0010 1 *  
11 0011 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
12 0012 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
13 0013 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
14 0014 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
15 0015 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
16 0016 1 * TRANSFERRED.  
17 0017 1 *  
18 0018 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
19 0019 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
20 0020 1 * CORPORATION.  
21 0021 1 *  
22 0022 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
23 0023 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
24 0024 1 *  
25 0025 1 *  
26 0026 1 *****  
27 0027 1  
28 0028 1  
29 0029 1 ++  
30 0030 1 FACILITY: FORTRAN SUPPORT LIBRARY  
31 0031 1  
32 0032 1 ABSTRACT:  
33 0033 1  
34 0034 1 This module is the run-time FORTRAN format compiler, FOR$$FMT_COMPIL.  
35 0035 1 It translates a format into the same form that the FORTRAN  
36 0036 1 compiler does. This module is adapted from the equivalent  
37 0037 1 compiler module, therefore changes in this module should be  
38 0038 1 evaluated to see if the compiler should be changed, and vice versa.  
39 0039 1  
40 0040 1 ENVIRONMENT: User access mode; AST re-entrant  
41 0041 1  
42 0042 1 AUTHOR: Peter Yuo, CREATION DATE: 07-June-77  
43 0043 1  
44 0044 1 MODIFIED BY:  
45 0045 1  
46 0046 1 Joel Clinkenbeard (FORTRAN IV-PLUS)  
47 0047 1 Steven B. Lionel (Run-Time Library)  
48 0048 1 Version 2 15-May-1979  
49 0049 1  
50 0050 1 EDIT HISTORY:  
51 0051 1  
52 0052 1 2-001 - Update to level of Version 2.0 FORTRAN compiler, including  
53 0053 1 FORTRAN-77 format codes. SBL 15-May-1979  
54 0054 1 2-002 - X is now the same as TR. SBL 2-Aug-1979  
55 0055 1 2-003 - Eliminate an extraneous RETURN expression. JBS 06-SEP-1979  
56 0056 1 2-004 - Allow sequences such as ".,.,.)" without error. SBL 18-Dec-1979  
57 0057 1 2-005 - Allow null characters in quoted literals and Hollerith literals.
```

FOR\$FMTC
2-006

FORTAN OBJECT TIME FORMAT COMPILER

B 10
16-Sep-1984 00:23:29
14-Sep-1984 12:31:59

VAX-11 Bliss-32 V4.0-742
[FORRTL.SRC]FORFMTC.B32;1

Page 2
(1)

:	58	0058	1	:	SPR 11-44210 SBL 1-March-1982
:	59	0059	1	:	2-006 - Allow W value to be zero; new extension for V4. Use prologue file.
:	60	0060	1	:	SBL 26-Apr-1983
:	61	0061	1	:	--
:	62	0062	1	:	

FOI
2-

00
00
00
00
00
00


```
64 0063 1 |
65 0064 1 | PROLOGUE FILE:
66 0065 1 |
67 0066 1 |
68 0067 1 | REQUIRE 'RTLIN:FORPROLOG';          ! FORTRAN definitions
69 0133 1 |
70 0134 1 |
71 0135 1 | LINKAGES:
72 0136 1 |
73 0137 1 |
74 0138 1 | LINKAGE
75 0139 1 |     CALL_G3 = CALL : GLOBAL (SAVVAL = 11, SAVTYP = 10, PTR = 9);
76 0140 1 |
77 0141 1 |
78 0142 1 | TABLE OF CONTENTS:
79 0143 1 |
80 0144 1 |
81 0145 1 | FORWARD ROUTINE
82 0146 1 |     FOR$FMT_COMPIL : NOVALUE,
83 0147 1 |     REDUCE : NOVALUE CALL_G3,
84 0148 1 |     DEFER : NOVALUE CALL_G3,
85 0149 1 |     UNDEFER : NOVALUE CALL_G3,
86 0150 1 |     NZERO : NOVALUE CALL_G3,
87 0151 1 |     NSAVE : NOVALUE CALL_G3,
88 0152 1 |     PUTBYT : NOVALUE CALL_G3,
89 0153 1 |     BYTSIZ;
90 0154 1 |
91 0155 1 |
92 0156 1 | MACROS:
93 0157 1 |
94 0158 1 |
95 0159 1 | MACRO
96 M 0160 1 |     ERROR (ERR_SYM) =
97 M 0161 1 |         (FOR$SIGNAL STO (FOR$K_SYNERRFOR);
98 M 0162 1 |         RETURN (0)) %,
99 M 0163 1 |     EXT_REG =
100 M 0164 1 |         EXTERNAL REGISTER
101 M 0165 1 |         SAVVAL: REF VECTOR[.LONG],
102 M 0166 1 |         SAVTYP: REF VECTOR[.LONG],
103 M 0167 1 |         PTR: REF VECTOR[.LONG] %,
104 M 0168 1 |     GC =
105 M 0169 1 |
106 M 0170 1 |         CH$RCHAR_A (FORMAT_PTR) %,
107 M 0171 1 |     GNB =
108 M 0172 1 |
109 M 0173 1 |         BEGIN
110 M 0174 1 |         FORMAT_PTR = CH$FIND_NOT_CH (K_MAX_LENGTH, .FORMAT_PTR, %C' ');
111 M 0175 1 |         IF CH$FAIL (.FORMAT_PTR)
112 M 0176 1 |         THEN
113 M 0177 1 |             ERROR (ERRFMTCHAR);
114 M 0178 1 |         BEGIN
115 M 0179 1 |         LOCAL
116 M 0180 1 |         C;
117 M 0181 1 |         C = CH$RCHAR_A (FORMAT_PTR);
118 M 0182 1 |         IF (.C GEQU %C'a') AND (.C LEQU %C'z')
119 M 0183 1 |         THEN
120 M 0184 1 |             .C = (%C'a' - %C'A')
```



```
121      ELSE
122      .C
123      END
124      END %;
125
126      !
127      ! EXTERNAL REFERENCES:
128      !
129      !
130      ! EXTERNAL ROUTINE
131      ! FOR$GET_VM,
132      ! FOR$FREE_VM : NOVALUE,
133      ! FOR$SIGNAL_STO : NOVALUE;
134      !
135      ! Get dynamic virtual memory
136      ! Free dynamic virtual memory
137      ! signal-stop FOR$_abcmnoxyz, given
138      !
139      ! (short) Fortran error number (FOR$_abcmnoxyz)
140      ! as a parameter
141
142      !
143      ! OWN STORAGE:
144      !
145      ! NONE
146      !
147      ! EQUATED SYMBOLS:
148      !
149      ! LITERAL
150      ! TRUE = 1,
151      ! K_FMT_BUF_INIT = 256,
152      ! K_MAX_LENGTH = 65535,
153      !
154      ! initial length (bytes) of format buffer
155      ! max. length of input character array
156      !
157      ! +
158      ! Define offsets into LOCAL VECTOR pointed to by GLOBAL register PTR
159      ! -
160      !
161      ! L_FDEFER = 0,
162      ! L_FCOUNT = 1,
163      ! L_PHASE = 2,
164      ! L_NEST = 3,
165      ! L_SIGN = 4,
166      ! L_NVAL = 5,
167      ! L_TYPE = 6,
168      ! L_NCHAR = 7,
169      ! A_FMT_BUF_BEG = 8,
170      ! L_CPRIME = 9,
171      ! L_FMT_BUF_SIZ = 10,
172      !
173      ! format code for deferred item
174      ! count of W, D, for deferred item
175      ! index to SAVVAL and SAVTYP
176      ! parenthesis nest level
177      ! non-zero if minus sign seen
178      ! value of numeric item
179      ! type of numeric item
180      ! character index within FMT_BUF
181      ! pointer to beginning of compiled output
182      ! previous character
183      ! current size (bytes) of dynamically allocated format buffer
184      !
185      ! +
186      ! Define size constants for the LOCAL structures
187      ! -
188      !
189      ! K_PTR_SIZ = 11,
190      ! K_SAVVAL_SIZ = 4,
191      ! K_SAVTYP_SIZ = 4,
192      ! K_PTR_OFFSET = K_SAVVAL_SIZ + K_SAVTYP_SIZ,
193      ! K_LOCAL_SIZ = K_PTR_OFFSET + K_PTR_SIZ;
194      !
195      ! No. of local variables pointed to by PTR
196      ! No. of longwords in SAVVAL
197      ! No. of longwords in SAVTYP
198      ! Offset into local storage
199      ! of PTR
200      ! Total size of LOCAL storage (longwords)
201
202      !
203      ! BIND
```



```
178 0242 1 ! CHARACTER CLASS TABLE
179 0243 1 !
180 0244 1 K CLASS_TAB MAX = 132, ! MAX. LEGAL CHARACTER (OUTSIDE OF STRING CONSTANT)
181 0245 1 CLASS = -UPLIT BYTE(
182 0246 1 1, 0, 0, 0, 0, 0, 0, 0, 000
183 0247 1 0, 0, 0, 0, 0, 0, 0, 0, 010
184 0248 1 0, 0, 0, 0, 0, 0, 0, 0, 020
185 0249 1 0, 0, 0, 0, 0, 0, 0, 0, 030
186 0250 1 0, 0, 0, 0, 0, 0, 0, 13, 040
187 0251 1 6, 7, 0, 3, 11, 2, 12, 8, 050
188 0252 1 5, 5, 5, 5, 5, 5, 5, 5, 060
189 0253 1 5, 5, 10, 0, 4, 0, 0, 0, 070
190 0254 1 0, 14, 15, 0, 16, 17, 18, 19, 100
191 0255 1 20, 21, 0, 0, 22, 0, 0, 23, 110
192 0256 1 24, 25, 0, 26, 27, 0, 0, 0, 120
193 0257 1 28, 0, 29) : VECTOR [, BYTE]; ! 130
194 0258 1
195 0259 1 BIND
196 0260 1 ! FORMAT CODES
197 0261 1 !
198 0262 1 TOPLVL = 1, ! Format reversion point
199 0263 1 LPAREN = 2, ! Left parenthesis
200 0264 1 RPAREN = 3, ! Right parenthesis
201 0265 1 ENDFMT = 4, ! End of format
202 0266 1 SLASH = 5, ! Slash
203 0267 1 DOLLAR = 6, ! Dollar sign
204 0268 1 COLON = 7, ! Colon
205 0269 1 SCODE = 9, ! S
206 0270 1 SPCODE = 10, ! SP
207 0271 1 SSCODE = 11, ! SS
208 0272 1 PCODE = 12, ! P
209 0273 1 TCODE = 13, ! T
210 0274 1 XCODE = 14, ! X
211 0275 1 HCODE = 15, ! H or quote
212 0276 1 BNCODE = 16, ! BN
213 0277 1 BZCODE = 17, ! BZ
214 0278 1 TLCODE = 18, ! TL
215 0279 1 TRCODE = 19, ! TR
216 0280 1 QCODE = 20, ! Q
217 0281 1 ACODE = 21, ! A
218 0282 1 LCODE = 22, ! L
219 0283 1 OCODE = 23, ! O
220 0284 1 ICODE = 24, ! I
221 0285 1 ZCODE = 25, ! Z
222 0286 1 FCODE = 30, ! F
223 0287 1 ECODE = 31, ! E
224 0288 1 GCODE = 32, ! G
225 0289 1 DCODE = 33, ! D
226 0290 1 IOZOFFSET = 3, ! Offset for Iw.m,Ow.m,Zw.m
227 0291 1 EGOFFSET = 3, ! Offset for E,G with Ee exponent
228 0292 1 OFFSET = 20; ! Offset to default A...D codes
229 0293 1
```

[illegible]


```

288      0351      2      FORMAT_PTR,      ! Address of last character from source
289      0352      2      FMTDAT = VECTOR [K_LOCAL_SIZ];      ! impure data for format processing
290      0353      2
291      0354      2      !+
292      0355      2      Bind names to LOCAL storage for this routine only.  Calls to other routines
293      0356      2      access these locations using .PTR[L_name].
294      0357      2      !-
295      0358      2
296      0359      2      BIND
297      0360      2      FDEFER = FMTDAT [K_PTR_OFFSET + L_FDEFER],      ! FORMAT CODE FOR DEFERRED ITEM
298      0361      2      FCOUNT = FMTDAT [K_PTR_OFFSET + L_FCOUNT],      ! COUNT OF W, D FOR DEFERRED ITEM
299      0362      2      PHASE = FMTDAT [K_PTR_OFFSET + L_PHASE],      ! INDEX TO SAVVAL AND SAVTYP
300      0363      2      NEST = FMTDAT [K_PTR_OFFSET + L_NEST],      ! PARENTHESIS NEST LEVEL
301      0364      2      SIGN = FMTDAT [K_PTR_OFFSET + L_SIGN],      ! -1 if neg, 1 if pos, 0 if no sign
302      0365      2      NVAL = FMTDAT [K_PTR_OFFSET + L_NVAL],      ! VALUE OF NUMERIC ITEM
303      0366      2      TYPE = FMTDAT [K_PTR_OFFSET + L_TYPE],      ! TYPE OF NUMERIC ITEM
304      0367      2      ! -1 = VARIABLE FORMAT EXPRESSION
305      0368      2      ! 0 = NOT PRESENT
306      0369      2      ! +1 = CONSTANT
307      0370      2      NCHAR = FMTDAT [K_PTR_OFFSET + L_NCHAR],      ! CHARACTER INDEX WITHIN FMT_BUF
308      0371      2      FMT_BUF_BEG = FMTDAT [K_PTR_OFFSET + A_FMT_BUF_BEG],      ! POINTER TO BEGINING OF COMPILED OUTPUT FORMAT BUFFER
309      0372      2
310      0373      2      CPRIME = FMTDAT [K_PTR_OFFSET + L_CPRIME],      ! PREVIOUS CHARACTER
311      0374      2      FMT_BUF_SIZ = FMTDAT [K_PTR_OFFSET + L_FMT_BUF_SIZ];
312      0375      2
313      0376      2      ! CURRENT ALLOCATION FOR DYNAMICALLY ALLOCATED FORMAT BUFFER
314      0377      2
315      0378      2      !+
316      0379      2      Setup GLOBAL registers to be passed to other routines
317      0380      2      !-
318      0381      2
319      0382      2      SAVVAL = FMTDAT [0];      ! Set pointer to value of N, W, D parameters
320      0383      2      SAVTYP = FMTDAT [K_SAVVAL_SIZ];      ! Set pointer to type of N, W, D parameters
321      0384      2      PTR = FMTDAT [K_PTR_OFFSET];      ! Set pointer to remainder of local storage
322      0385      2      ! ACTUALLY PROCESS THE FORMAT SPECIFICATION
323      0386      2      ! Clear LOCAL storage, and allocate initial format buffer
324      0387      2
325      0388      2      FILL VAL (0, K_LOCAL_SIZ, FMTDAT);
326      0389      2      FMT_BUF_BEG = FOR$$GET_VM (K_FMT_BUF_INIT);
327      0390      2      FMT_BUF_SIZ = K_FMT_BUF_INIT;
328      0391      2      CPRIME = '(';
329      0392      2      FORMAT_PTR = CH$PTR (.FORMAT);
330      0393      2      FORMAT_PTR = CH$FIND_NOT_CH (K_MAX_LENGTH, .FORMAT_PTR, %C' ');
331      0394      2
332      0395      2      IF CH$FAIL (.FORMAT_PTR) OR CH$RCHAR_A (FORMAT_PTR) NEQ %C'('
333      0396      2      THEN
334      0397      2          ERROR (ERRMISSDLM)
335      0398      2      ELSE
336      0399      2          BEGIN
337      0400      2
338      0401      2          WHILE 1 DO
339      0402      2              BEGIN
340      0403      2                  CHAR = GNB;      ! Get next non-blank
341      0404      2
342      0405      2                  IF .CHAR GTRU K_CLASS_TAB_MAX THEN ERROR (ERRFMTCHAR);
343      0406      2
344      0407      2                  CASE .CLASS [.CHAR] FROM 0 TO 29 OF

```

```

: 345      0408 4      SET
: 346      0409 4
: 347      0410 4      [0] :
: 348      0411 4          : 0 - INVALID CHARACTER
: 349      0412 4          :
: 350      0413 4          : ERROR (ERRFMTCHAR);
: 351      0414 4
: 352      0415 4      [1] :
: 353      0416 4          : 1 - NULL CHARACTER
: 354      0417 4          :
: 355      0418 4          : ERROR (ERRFMTRPAR);
: 356      0419 4
: 357      0420 4      [2] :
: 358      0421 4          : 2 - MINUS SIGN
: 359      0422 4          :
: 360      0423 5          : BEGIN
: 361      0424 5
: 362      0425 5          : IF .SIGN NEQ 0 OR .TYPE NEQ 0 THEN ERROR (ERRFMTCHAR);
: 363      0426 5
: 364      0427 5          : SIGN = -1;
: 365      0428 4          : END;
: 366      0429 4
: 367      0430 4      [3] :
: 368      0431 4          : 3 - PLUS SIGN
: 369      0432 4          :
: 370      0433 5          : BEGIN
: 371      0434 5
: 372      0435 5          : IF .SIGN NEQ 0 OR .TYPE NEQ 0 THEN ERROR (ERRFMTCHAR);
: 373      0436 5
: 374      0437 5          : SIGN = 1;
: 375      0438 4          : END;
: 376      0439 4
: 377      0440 4      [4] :
: 378      0441 4          : 4 - LEFT ANGLE BRACKET
: 379      0442 4          :
: 380      0443 4          : ERROR (ERRFMTCHAR);
: 381      0444 4
: 382      0445 4      [5] :
: 383      0446 4          : 5 - DIGIT
: 384      0447 4          :
: 385      0448 5          : BEGIN
: 386      0449 5          : TYPE = 1;
: 387      0450 5          : NVAL = .NVAL*10 + .CHAR - '0';
: 388      0451 4          : END;
: 389      0452 4
: 390      0453 4      [6] :
: 391      0454 4          : 6 - LEFT PARENTHESIS
: 392      0455 4          :
: 393      0456 5          : BEGIN
: 394      0457 5          : NZERO ();
: 395      0458 5          : NSAVE ();
: 396      0459 5
: 397      0460 5          : IF .NEST EQL 0 THEN PUTBYT (TOPLVL);
: 398      0461 5
: 399      0462 5          : IF (NEST = .NEST + 1) GTR 8 THEN ERROR (ERRFMTNEST);
: 400      0463 5
: 401      0464 5          : REDUCE (LPAREN);
```



```

402      0465 4      END;
403      0466 4
404      0467 4
405      0468 4
406      0469 4
407      0470 5
408      0471 5
409      0472 5
410      0473 5
411      0474 5
412      0475 5
413      0476 5
414      0477 5
415      0478 5
416      0479 5 !
417      0480 5
418      0481 5
419      0482 5
420      0483 5
421      0484 5
422      0485 5
423      0486 4
424      0487 4
425      0488 4
426      0489 4
427      0490 4
428      0491 5
429      0492 5
430      0493 5
431      0494 4
432      0495 4
433      0496 4
434      0497 4
435      0498 4
436      0499 5
437      0500 5
438      0501 5
439      0502 4
440      0503 4
441      0504 4
442      0505 4
443      0506 4
444      0507 5
445      0508 5
446      0509 5
447      0510 4
448      0511 4
449      0512 4
450      0513 4
451      0514 4
452      0515 5
453      0516 5
454      0517 5
455      0518 5
456      0519 5
457      0520 5
458      0521 5 !

      [7] : 7 - RIGHT PARENTHESIS
      BEGIN
      +
      When the VAX-11 FORTRAN compiler sees the sequence ")",
      it issues a warning message and otherwise ignores the
      extra delimiter. A deliberate decision was made for
      release 2 to ignore this occurrence entirely in the
      run-time format compiler.
      -
      IF .CPRIME EQL ',' THEN ERROR (ERRFMXTCOM);
      UNDEFER ();
      IF (NEST = .NEST - 1) LSS 0 THEN EXITLOOP;
      PUTBYT (RPAREN);
      END;

      [8] : 8 - SLASH
      BEGIN
      UNDEFER ();
      PUTBYT (SLASH);
      END;

      [9] : 9 - DOLLAR SIGN
      BEGIN
      UNDEFER ();
      PUTBYT (DOLLAR);
      END;

      [10] : 10 - COLON
      BEGIN
      UNDEFER ();
      PUTBYT (COLON);
      END;

      [11] : 11 - COMMA
      BEGIN
      +
      The sequence "(", " or "(," is ignored here. See comment
      under RIGHT PARENTHESIS.
      -
      IF .CPRIME EQL ',' OR .CPRIME EQL '(' THEN ERROR (ERRFMXTCOM);
```

```

459 0522 5
460 0523 5
461 0524 4
462 0525 4
463 0526 4
464 0527 4
465 0528 4
466 0529 5
467 0530 5
468 0531 5
469 0532 5
470 0533 5
471 0534 5
472 0535 5
473 0536 5
474 0537 5
475 0538 4
476 0539 4
477 0540 4
478 0541 4
479 0542 4
480 0543 5
481 0544 5
482 0545 5
483 0546 5
484 0547 5
485 0548 5
486 0549 5
487 0550 5
488 0551 5
489 0552 6
490 0553 6
491 0554 6
492 0555 7
493 0556 7
494 0557 7
495 0558 7
496 0559 6
497 0560 6
498 0561 6
499 0562 6
500 0563 5
501 0564 5
502 0565 5
503 0566 5
504 0567 5
505 0568 5
506 0569 5
507 0570 5
508 0571 5
509 0572 5
510 0573 5
511 0574 5
512 0575 6
513 0576 6
514 0577 6
515 0578 6

      UNDEFER ();
      END;

[12] : 12 - DECIMAL POINT
      BEGIN
      IF .TYPE EQL 0 THEN ERROR (ERRFMTNUMB);
      IF .SIGN NEQ 0 THEN ERROR (ERRFMTRNGE);
      IF .FCOUNT LSS 2 OR .PHASE NEQ 1 THEN ERROR (ERRFMTCHAR);
      NSAVE ();
      END;

[13] : 13 - QUOTE
      BEGIN
      LOCAL
      P;

      UNDEFER ();
      P = .FORMAT_PTR;

      DO
      BEGIN
      DO
      BEGIN
      CHAR = GC;          ! Get next character
      NVAL = .NVAL + 1;
      END
      WHILE .CHAR NEQ ''';
      CHAR = GC;
      END
      WHILE .CHAR EQL ''';
      FORMAT_PTR = .P;
      IF (NVAL = P = .NVAL - 1) EQL 0 THEN ERROR (ERRZLSTR);
      TYPE = 1;
      PHASE = 1;
      NSAVE ();
      REDUCE (HCODE);
      DECR I FROM .P TO 1 DO
      BEGIN
      IF (CHAR = GC) EQL '''' THEN GC;
```



```

: 516      0579 6      PUTBYT (.CHAR);
: 517      0580 5      END;
: 518      0581 5
: 519      0582 5      CHAR = GC;
: 520      0583 4      END;
: 521      0584 4
: 522      0585 4      [14] :
: 523      0586 4      | 14 - LETTER A
: 524      0587 4      |
: 525      0588 4      | DEFER (ACODE, 1);
: 526      0589 4
: 527      0590 4      [15] :
: 528      0591 4      | 15 - Letter B
: 529      0592 4      |
: 530      0593 5      | BEGIN
: 531      0594 5      | UNDEFER ();
: 532      0595 5
: 533      0596 5      | SELECTONE (CHAR = GNB) OF
: 534      0597 5      | SET
: 535      0598 5
: 536      0599 5      | ['N'] :
: 537      0600 5      | PUTBYT (BNCODE);
: 538      0601 5
: 539      0602 5      | ['Z'] :
: 540      0603 5      | PUTBYT (BZCODE);
: 541      0604 5
: 542      0605 5      | [OTHERWISE] :
: 543      0606 6      | BEGIN
: 544      0607 6      | ERROR (ERRFMTCHAR);
: 545      0608 5      | END;
: 546      0609 5      | TES;
: 547      0610 5
: 548      0611 4      | END;
: 549      0612 4
: 550      0613 4      [16] :
: 551      0614 4      | 16 - LETTER D
: 552      0615 4      |
: 553      0616 4      | DEFER (DCODE, 2);
: 554      0617 4
: 555      0618 4      [17] :
: 556      0619 4      | 17 - LETTER E
: 557      0620 4      |
: 558      0621 4      |
: 559      0622 4      | +
: 560      0623 4      | If the third parameter of an edit type that allows four
: 561      0624 4      | parameters has been seen, then E is an exponent marker,
: 562      0625 4      | otherwise an edit specifier.
: 563      0626 4      | -
: 564      0627 4
: 565      0628 4      | IF .PHASE EQL 2 AND .FCOUNT EQL 3 THEN NSAVE () ELSE DEFER (ECODE, 3);
: 566      0629 4
: 567      0630 4      [18] :
: 568      0631 4      | 18 - LETTER F
: 569      0632 4      |
: 570      0633 4      | DEFER (FCODE, 2);
: 571      0634 4
: 572      0635 4      [19] :
```

```

: 573      0636      4      : 19 - LETTER G
: 574      0637      4      :
: 575      0638      4      DEFER (GCODE, 3);
: 576      0639      4      :
: 577      0640      4      [20] :
: 578      0641      4      : 20 - LETTER H
: 579      0642      4      :
: 580      0643      5      BEGIN
: 581      0644      5      :
: 582      0645      5      LOCAL
: 583      0646      5      P;
: 584      0647      5      :
: 585      0648      5      NZERO ();
: 586      0649      5      :
: 587      0650      5      IF .TYPE LSS 0 THEN ERROR (ERRFMTCHAR);
: 588      0651      5      :
: 589      0652      5      IF .TYPE EQL 0 THEN (NVAL = 1; TYPE = 1);
: 590      0653      5      :
: 591      0654      5      IF (P = .NVAL) EQL 0 THEN ERROR (ERRZLSTR);
: 592      0655      5      :
: 593      0656      5      PHASE = 1;
: 594      0657      5      NSAVE ();
: 595      0658      5      REDUCE (HCODE);
: 596      0659      5      :
: 597      0660      5      DECR I FROM .P TO 1 DO
: 598      0661      6      BEGIN
: 599      0662      6      :
: 600      0663      6      CHAR = GC;
: 601      0664      6      :
: 602      0665      6      PUTBYT (.CHAR);
: 603      0666      5      END;
: 604      0667      5      :
: 605      0668      5      CHAR = 0;
: 606      0669      4      END;
: 607      0670      4      :
: 608      0671      4      [21] :
: 609      0672      4      : 21 - LETTER I
: 610      0673      4      :
: 611      0674      4      DEFER (ICODE, 2);
: 612      0675      4      :
: 613      0676      4      [22] :
: 614      0677      4      : 22 - LETTER L
: 615      0678      4      :
: 616      0679      4      DEFER (LCODE, 1);
: 617      0680      4      :
: 618      0681      4      [23] :
: 619      0682      4      : 23 - LETTER O
: 620      0683      4      :
: 621      0684      4      DEFER (OCODE, 2);
: 622      0685      4      :
: 623      0686      4      [24] :
: 624      0687      4      : 24 - LETTER P
: 625      0688      4      :
: 626      0689      5      BEGIN
: 627      0690      5      NZERO ();
: 628      0691      5      :
: 629      0692      5      IF .TYPE EQL 0
```



```

: 630      0693 5      THEN
: 631      0694 6      BEGIN
: 632      0695 6
: 633      0696 6      IF .SIGN NEQ 0 THEN ERROR (ERRFMTNUMB);
: 634      0697 6
: 635      0698 5      END;
: 636      0699 5
: 637      0700 5      IF .SIGN LSS 0 THEN NVAL = -.NVAL;
: 638      0701 5
: 639      0702 5      SIGN = 0;
: 640      0703 5      PHASE = 1;
: 641      0704 5      NSAVE ();
: 642      0705 5      REDUCE (PCODE);
: 643      0706 4      END;
: 644      0707 4
: 645      0708 4      [25] :
: 646      0709 4      | 25 - LETTER Q
: 647      0710 4      |
: 648      0711 5      BEGIN
: 649      0712 5      UNDEFER ();
: 650      0713 5      PUTBYT (QCODE);
: 651      0714 4      END;
: 652      0715 4
: 653      0716 4      [26] :
: 654      0717 4      | 26 - Letter S
: 655      0718 4      |
: 656      0719 5      BEGIN
: 657      0720 5      UNDEFER ();
: 658      0721 5
: 659      0722 5      SELECTONE (CHAR = GNB) OF
: 660      0723 5      SET
: 661      0724 5      ['P'] :
: 662      0725 5      | PUTBYT (SPCODE);
: 663      0726 5
: 664      0727 5      ['S'] :
: 665      0728 5      | PUTBYT (SSCODE);
: 666      0729 5
: 667      0730 5      [OTHERWISE] :
: 668      0731 5      | BEGIN
: 669      0732 6      | PUTBYT (SCODE);
: 670      0733 6      | FORMAT_PTR = .FORMAT_PTR - 1;
: 671      0734 6      | CHAR = 'S';
: 672      0735 6      | END;
: 673      0736 5      TES;
: 674      0737 5
: 675      0738 5      END;
: 676      0739 4
: 677      0740 4      [27] :
: 678      0741 4      | 27 - LETTER T
: 679      0742 4      |
: 680      0743 4      | BEGIN
: 681      0744 5      | UNDEFER ();
: 682      0745 5
: 683      0746 5      SELECTONE (CHAR = GNB) OF
: 684      0747 5      SET
: 685      0748 5
: 686      0749 5
```

```

: 687      0750 5      ['L'] :
: 688      0751 5      DEFER (TLCODE, 1);
: 689      0752 5
: 690      0753 5      ['R'] :
: 691      0754 5      DEFER (TRCODE, 1);
: 692      0755 5
: 693      0756 5      [OTHERWISE] :
: 694      0757 6      BEGIN
: 695      0758 6      DEFER (TCODE, 1);
: 696      0759 6      FORMAT_PTR = .FORMAT_PTR - 1;
: 697      0760 6      CHAR = 'T';
: 698      0761 5      END;
: 699      0762 5      TES;
: 700      0763 5
: 701      0764 4      END;
: 702      0765 4
: 703      0766 4      [28] :
: 704      0767 4      ! 28 - LETTER X
: 705      0768 4      !
: 706      0769 5      BEGIN
: 707      0770 5      NZERO ();
: 708      0771 5
: 709      0772 5      IF .TYPE EQL 0
: 710      0773 5      THEN
: 711      0774 6      BEGIN
: 712      0775 6      TYPE = 1;
: 713      0776 6      NVAL = 1;
: 714      0777 5      END;
: 715      0778 5
: 716      0779 5      PHASE = 1;
: 717      0780 5      NSAVE ();
: 718      0781 5      REDUCE (TRCODE);
: 719      0782 5      ! X is same as TR
: 720      0783 4      END;      ! Old X is no longer used.
: 721      0784 4
: 722      0785 4      [29] :
: 723      0786 4      ! 29 - LETTER Z
: 724      0787 4      !
: 725      0788 4      DEFER (ZCODE, 2)
: 726      0789 4      TES;
: 727      0790 4
: 728      0791 4      CPRIME = .CHAR;
: 729      0792 3      END;
: 730      0793 3
: 731      0794 3      !+
: 732      0795 3      ! Put end of format code.
: 733      0796 3      ! Then return size and location of format buffer.
: 734      0797 3      !-
: 735      0798 3
: 736      0799 3      PUTBYT (ENDFMT);
: 737      0800 3      ALLOCATED_LEN [0] = .FMT_BUF_SIZ;
: 738      0801 3      ALLOCATED_ADR [0] = .FMT_BUF_BEG;
: 739      0802 2      END;
: 740      0803 2
: 741      0804 1      END;
```



```

.ENTRY  FOR$$FMT COMPIL, Save R2,R3,R4,R5,R6,R7,R8,-; 0294
MOVAB   R9,R10,RT1
MOVAB   PUTBYT, R8
MOVAB   NSAVE, R7
MOVAB   UNDEFER, R6
MOVAB   -76(SP), SP
MOVAB   FMTDAT, SAVVAL

```

58	0000V	CF	9E	00002
57	0000V	CF	9E	00007
56	0000V	CF	9E	0000C
5E	B4	AE	9E	00011
5B		6E	9E	00015

004C	8F	00	5A	10	AE	9E	00018	MOVAB	FMTDAT+16, SAVTYP	:	0383
			59	20	AE	9E	0001C	MOVAB	FMTDAT+32, PTR	:	0384
			6E		00	2C	00020	MOVC5	#0, (SP), #0, #76, FMTDAT	:	0388
					6E		00027			:	
			7E	0100	8F	3C	00028	MOVZWL	#256, -(SP)	:	0389
		00000000G	00		01	FB	0002D	CALLS	#1, FOR\$GET_VM	:	
		40	AE		50	D0	00034	MOVL	R0, FMT_BUF_BEG	:	
		48	AE	0100	8F	3C	00038	MOVZWL	#256, FMT_BUF_SIZ	:	0390
		44	AE		28	D0	0003E	MOVL	#40, CPRIME	:	0391
			54	04	AC	D0	00042	MOVL	FORMAT, FORMAT_PTR	:	0392
64		FFFF	8F		20	3B	00046	SKPC	#32, #65535, (FORMAT_PTR)	:	0393
					02	12	0004C	BNEQ	1\$:	
			54		51	D4	0004E	CLRL	R1	:	
					51	D0	00050	MOVL	R1, FORMAT_PTR	:	
			50		7E	13	00053	BEQL	6\$:	0395
			28		84	9A	00055	MOVZBL	(FORMAT_PTR)+, R0	:	
					50	91	00058	CMPB	R0, #40	:	
					76	12	0005B	BNEQ	6\$:	
64		FFFF	8F		20	3B	0005D	SKPC	#32, #65535, (FORMAT_PTR)	:	0403
					02	12	00063	BNEQ	3\$:	
			54		51	D4	00065	CLRL	R1	:	
					51	D0	00067	MOVL	R1, FORMAT_PTR	:	
					67	13	0006A	BEQL	6\$:	
			50		84	9A	0006C	MOVZBL	(FORMAT_PTR)+, C	:	
		00000061	8F		50	D1	0006F	CMPL	C, #97	:	
					0C	1F	00076	BLSSU	4\$:	
		0000007A	8F		50	D1	00078	CMPL	C, #122	:	
					03	1A	0007F	BGTRU	4\$:	
			50		20	C2	00081	SUBL2	#32, R0	:	
			52		50	D0	00084	MOVL	C, CHAR	:	
		00000084	8F		52	D1	00087	CMPL	CHAR, #132	:	0405
					43	1A	0008E	BGTRU	6\$:	
					43	1A	0008E	CASEB	CLASS[CHAR], #0, #29	:	0407
					8F	00090		.WORD	66\$-5\$,-	:	
						00097	5\$:		66\$-5\$,-	:	
						0009F			7\$-5\$,-	:	
						000A7			8\$-5\$,-	:	
						000AF			66\$-5\$,-	:	
						000B7			9\$-5\$,-	:	
						000BF			10\$-5\$,-	:	
						000C7			12\$-5\$,-	:	
						000CF			14\$-5\$,-	:	
									15\$-5\$,-	:	
									16\$-5\$,-	:	
									18\$-5\$,-	:	
									20\$-5\$,-	:	
									23\$-5\$,-	:	
									29\$-5\$,-	:	
									30\$-5\$,-	:	
									35\$-5\$,-	:	
									36\$-5\$,-	:	
									40\$-5\$,-	:	
									41\$-5\$,-	:	
									43\$-5\$,-	:	
									50\$-5\$,-	:	
									51\$-5\$,-	:	
									52\$-5\$,-	:	

004F	1D	00	FF10	CF42	8F	00090	
0095	003F	0290		0290		00097	5\$:
00BA	0070	005F		0290		0009F	
0135	00B2	00AB		00A4		000A7	
01A0	012F	00DB		00BF		000AF	
01F1	019A	0183		017D		000B7	
027E	01EB	01E5		01A6		000BF	
	0227	0220		01F8		000C7	
		0306		02E4		000CF	

						54\$-5\$,-		
						57\$-5\$,-		
						58\$-5\$,-		
						64\$-5\$,-		
						72\$-5\$,-		
						75\$-5\$		
						66\$		0413
						SIGN		0425
						6\$		
						TYPE		
						6\$		
						#1, SIGN		0427
						19\$		0407
						SIGN		0435
						6\$		
						TYPE		
						6\$		
						#1, SIGN		0437
						19\$		0407
						#1, TYPE		0449
						#10, NVAL, RO		0450
						-48(CHAR)[RO], NVAL		
						19\$		0407
						#0, NZERO		0457
						#0, NSAVE		0458
						NEST		0460
						11\$		
						#1		
						#1, PUTBYT		
						#1, NEST, RO		0462
						RO, NEST		
						RO, #8		
						6\$		
						#2		0464
						74\$		
						#0, UNDEFER		0481
						NEST		0483
						13\$		
						78\$		
						#3		0485
						17\$		
						#0, UNDEFER		0492
						#5		0493
						17\$		
						#0, UNDEFER		0500
						#6		0501
						17\$		
						#0, UNDEFER		0508
						#7		0509
						62\$		
						#0, UNDEFER		0523
						28\$		0407
						TYPE		0531
						21\$		
						SIGN		0533
						21\$		
						FCOUNT, #2		0535

		03	18	00164	BGEQ	22\$:	
		01BE	31	00166	BRW	66\$:	
	01	28	AE	D1	00169	21\$:	CMPL	PHASE, #1
			F7	12	0016D	22\$:	BNEQ	21\$
			00B4	31	0016F		BRW	37\$
	66		00	FB	00172	23\$:	CALLS	#0, UNDEFER
	53		54	D0	00175		MOVL	FORMAT_PTR, P
	52		84	9A	00178	24\$:	MOVZBL	(FORMAT_PTR)+, CHAR
		34	AE	D6	0017B		INCL	NVAL
	27		52	D1	0017E		CMPL	CHAR, #39
			F5	12	00181		BNEQ	24\$
	52		84	9A	00183		MOVZBL	(FORMAT_PTR)+, CHAR
	27		52	D1	00186		CMPL	CHAR, #39
			ED	13	00189		BEQL	24\$
	54		53	D0	0018B		MOVL	P, FORMAT_PTR
53	34	AE	01	C3	0018E		SUBL3	#1, NVAL, -P
	34	AE	53	D0	00193		MOVL	P, NVAL
			CD	13	00197		BEQL	21\$
	38	AE	01	D0	00199		MOVL	#1, TYPE
	28	AE	01	D0	0019D		MOVL	#1, PHASE
	67		00	FB	001A1		CALLS	#0, NSAVE
			0F	DD	001A4		PUSHL	#15
	0000V	CF	01	FB	001A6		CALLS	#1, REDUCE
			53	D6	001AB		INCL	I
			0F	11	001AD		BRB	27\$
	52		84	9A	001AF	25\$:	MOVZBL	(FORMAT_PTR)+, CHAR
	27		52	D1	001B2		CMPL	CHAR, #39
			02	12	001B5		BNEQ	26\$
			54	D6	001B7		INCL	FORMAT_PTR
			52	DD	001B9	26\$:	PUSHL	CHAR
	68		01	FB	001BB		CALLS	#1, PUTBYT
	EE		53	F5	001BE	27\$:	SOBGTR	I, 25\$
	52		84	9A	001C1		MOVZBL	(FORMAT_PTR)+, CHAR
			63	11	001C4	28\$:	BRB	38\$
			01	DD	001C6	29\$:	PUSHL	#1
			15	DD	001C8		PUSHL	#21
			6F	11	001CA		BRB	42\$
			00	FB	001CC	30\$:	CALLS	#0, UNDEFER
64	FFFF	8F	20	3B	001CF		SKPC	#32, #65535, (FORMAT_PTR)
			02	12	001D5		BNEQ	31\$
			51	D4	001D7		CLRL	R1
	54		51	D0	001D9	31\$:	MOVL	R1, FORMAT_PTR
			69	13	001DC		BEQL	44\$
	50		84	9A	001DE		MOVZBL	(FORMAT_PTR)+, C
	00000061	8F	50	D1	001E1		CMPL	C, #97
			0C	1F	001E8		BLSSU	32\$
	0000007A	8F	50	D1	001EA		CMPL	C, #122
			03	1A	001F1		BGTRU	32\$
	50		20	C2	001F3		SUBL2	#32, R0
	52		50	D0	001F6	32\$:	MOVL	C, CHAR
	0000004E	8F	52	D1	001F9		CMPL	CHAR, #78
			04	12	00200		BNEQ	33\$
			10	DD	00202		PUSHL	#16
			0B	11	00204		BRB	34\$
	0000005A	8F	52	D1	00206	33\$:	CMPL	CHAR, #90
			38	12	0020D		BNEQ	44\$
			11	DD	0020F		PUSHL	#17

		00EF	31	00211	34\$:	BRW	62\$:	
		02	DD	00214	35\$:	PUSHL	#2	:	0616
		21	DD	00216		PUSHL	#33	:	
		72	11	00218		BRB	53\$:	
	02	28	AE	D1 0021A	36\$:	CMPL	PHASE, #2	:	0628
		0B	12	0021E		BNEQ	39\$:	
	03	24	AE	D1 00220		CMPL	FCOUNT, #3	:	
		05	12	00224		BNEQ	39\$:	
	67		00	FB 00226	37\$:	CALLS	#0, NSAVE	:	
		4E	11	00229	38\$:	BRB	49\$:	
		03	DD	0022B	39\$:	PUSHL	#3	:	
		1F	DD	0022D		PUSHL	#31	:	
		5B	11	0022F		BRB	53\$:	
		02	DD	00231	40\$:	PUSHL	#2	:	0633
		1E	DD	00233		PUSHL	#30	:	
		55	11	00235		BRB	53\$:	
		03	DD	00237	41\$:	PUSHL	#3	:	0638
		20	DD	00239		PUSHL	#32	:	
		4F	11	0023B	42\$:	BRB	53\$:	
0000V	CF		00	FB 0023D	43\$:	CALLS	#0, NZERO	:	0648
		38	AE	D5 00242		TSTL	TYPE	:	0650
		03	18	00245		BGEQ	45\$:	
		00DD	31	00247	44\$:	BRW	66\$:	
		08	12	0024A	45\$:	BNEQ	46\$:	0652
34	AE		01	D0 0024C		MOVL	#1, NVAL	:	
38	AE		01	D0 00250		MOVL	#1, TYPE	:	
	53	34	AE	D0 00254	46\$:	MOVL	NVAL, P	:	0654
			ED	13 00258		BEQL	44\$:	
28	AE		01	D0 0025A		MOVL	#1, PHASE	:	0656
	67		00	FB 0025E		CALLS	#0, NSAVE	:	0657
			0F	DD 00261		PUSHL	#15	:	0658
0000V	CF		01	FB 00263		CALLS	#1, REDUCE	:	
			53	D6 00268		INCL	I	:	0660
			08	11 0026A		BRB	48\$:	
	52		84	9A 0026C	47\$:	MOVZBL	(FORMAT_PTR)+, CHAR	:	0663
			52	DD 0026F		PUSHL	CHAR	:	0665
	68		01	FB 00271		CALLS	#1, PUTBYT	:	
	F5		53	F5 00274	48\$:	SOBGTR	I, 47\$:	0660
			52	D4 00277		CLRL	CHAR	:	0668
		012A	31	00279	49\$:	BRW	77\$:	0407
			02	DD 0027C	50\$:	PUSHL	#2	:	0674
			18	DD 0027E		PUSHL	#24	:	
			0A	11 00280		BRB	53\$:	
			01	DD 00282	51\$:	PUSHL	#1	:	0679
			16	DD 00284		PUSHL	#22	:	
			04	11 00286		BRB	53\$:	
			02	DD 00288	52\$:	PUSHL	#2	:	0684
			17	DD 0028A		PUSHL	#23	:	
		0112	31	0028C	53\$:	BRW	76\$:	
0000V	CF		00	FB 0028F	54\$:	CALLS	#0, NZERO	:	0690
		38	AE	D5 00294		TSTL	TYPE	:	0692
			05	12 00297		BNEQ	55\$:	
		30	AE	D5 00299		TSTL	SIGN	:	0696
			A9	12 0029C		BNEQ	44\$:	
		30	AE	D5 0029E	55\$:	TSTL	SIGN	:	0700
			05	18 002A1		BGEQ	56\$:	
34	AE	34	AE	CE 002A3		MNEGL	NVAL, NVAL	:	

		28	AE	30	AE	D4	002A8	56\$:	CLRL	SIGN		0702
			67		01	D0	002AB		MOVL	#1, PHASE		0703
					00	FB	002AF		CALLS	#0, NSAVE		0704
					0C	DD	002B2		PUSHL	#12		0705
					00DF	31	002B4		BRW	74\$		
		66			00	FB	002B7	57\$:	CALLS	#0, UNDEFER		0712
					14	DD	002BA		PUSHL	#20		0713
					45	11	002BC		BRB	62\$		
		66			00	FB	002BE	58\$:	CALLS	#0, UNDEFER		0720
64	FFFF	8F			20	3B	002C1		SKPC	#32, #65535, (FORMAT_PTR)		0722
					02	12	002C7		BNEQ	59\$		
					51	D4	002C9		CLRL	R1		
		54			51	D0	002CB	59\$:	MOVL	R1, FORMAT_PTR		
					57	13	002CE		BEQL	66\$		
		50			84	9A	002D0		MOVZBL	(FORMAT_PTR)+, C		
	00000061	8F			50	D1	002D3		CMPL	C, #97		
					0C	1F	002DA		BLSSU	60\$		
	0000007A	8F			50	D1	002DC		CMPL	C, #122		
					03	1A	002E3		BGTRU	60\$		
		50			20	C2	002E5		SUBL2	#32, R0		
		52			50	D0	002E8	60\$:	MOVL	C, CHAR		
	00000050	8F			52	D1	002EB		CMPL	CHAR, #80		0725
					04	12	002F2		BNEQ	61\$		
					0A	DD	002F4		PUSHL	#10		0726
					0B	11	002F6		BRB	62\$		
	00000053	8F			52	D1	002F8	61\$:	CMPL	CHAR, #83		0728
					07	12	002FF		BNEQ	63\$		
					0B	DD	00301		PUSHL	#11		0729
		68			01	FB	00303	62\$:	CALLS	#1, PUTBYT		
					71	11	00306		BRB	71\$		
					09	DD	00308	63\$:	PUSHL	#9		0733
		68			01	FB	0030A		CALLS	#1, PUTBYT		
					54	D7	0030D		DECL	FORMAT_PTR		0734
		52		53	8F	9A	0030F		MOVZBL	#83, CHAR		0735
					64	11	00313		BRB	71\$		0407
		66			00	FB	00315	64\$:	CALLS	#0, UNDEFER		0745
64	FFFF	8F			20	3B	00318		SKPC	#32, #65535, (FORMAT_PTR)		0747
					02	12	0031E		BNEQ	65\$		
					51	D4	00320		CLRL	R1		
		54			51	D0	00322	65\$:	MOVL	R1, FORMAT_PTR		
					0A	12	00325		BNEQ	67\$		
					3E	DD	00327	66\$:	PUSHL	#62		
	00000000G	00			01	FB	00329		CALLS	#1, FOR\$\$SIGNAL_STO		
					04	00330		RET				
		50			84	9A	00331	67\$:	MOVZBL	(FORMAT_PTR)+, C		
	00000061	8F			50	D1	00334		CMPL	C, #97		
					0C	1F	0033B		BLSSU	68\$		
	0000007A	8F			50	D1	0033D		CMPL	C, #122		
					03	1A	00344		BGTRU	68\$		
		50			20	C2	00346		SUBL2	#32, R0		
		52			50	D0	00349	68\$:	MOVL	C, CHAR		
	0000004C	8F			52	D1	0034C		CMPL	CHAR, #76		0750
					06	12	00353		BNEQ	69\$		
					01	DD	00355		PUSHL	#1		0751
					12	DD	00357		PUSHL	#18		
					46	11	00359		BRB	76\$		
	00000052	8F			52	D1	0035B	69\$:	CMPL	CHAR, #82		0753

			06	12	00362	BNEQ	70\$		
			01	DD	00364	PUSHL	#1	0754	
			13	DD	00366	PUSHL	#19		
			37	11	00368	BRB	76\$		
			01	DD	0036A	70\$: PUSHL	#1	0758	
			0D	DD	0036C	PUSHL	#13		
0000V	CF		02	FB	0036E	CALLS	#2, DEFER		
			54	D7	00373	DECL	FORMAT_PTR	0759	
	52	54	8F	9A	00375	MOVZBL	#84, CHAR	0760	
			2B	11	00379	71\$: BRB	77\$	0407	
0000V	CF		00	FB	0037B	72\$: CALLS	#0, NZERO	0770	
		38	AE	D5	00380	TSTL	TYPE	0772	
			08	12	00383	BNEQ	73\$		
38	AE		01	D0	00385	MOVL	#1, TYPE	0775	
34	AE		01	D0	00389	MOVL	#1, NVAL	0776	
28	AE		01	D0	0038D	73\$: MOVL	#1, PHASE	0779	
67			00	FB	00391	CALLS	#0, NSAVE	0780	
			13	DD	00394	PUSHL	#19	0781	
0000V	CF		01	FB	00396	74\$: CALLS	#1, REDUCE		
			09	11	0039B	BRB	77\$	0407	
			02	DD	0039D	75\$: PUSHL	#2	0788	
			19	DD	0039F	PUSHL	#25		
0000V	CF		02	FB	003A1	76\$: CALLS	#2, DEFER		
44	AE		52	D0	003A6	77\$: MOVL	CHAR, CPRIME	0791	
		FCB0	31	003AA	BRW	2\$		0401	
			04	DD	003AD	78\$: PUSHL	#4	0799	
	68		01	FB	003AF	CALLS	#1, PUTBYT		
08	BC	48	AE	B0	003B2	MOVW	FMT_BUF_SIZ, @ALLOCATED_LEN	0800	
0C	BC	40	AE	D0	003B7	MOVL	FMT_BUF_BEG, @ALLOCATED_ADR	0801	
			04	003BC	RET			0804	

; Routine Size: 957 bytes, Routine Base: _FOR\$CODE + 005B

; 742 0805 1

```

744 0806 1 ROUTINE REDUCE (C) : CALL_G3 NOVALUE =
745 0807 1
746 0808 1 ++
747 0809 1 FUNCTIONAL DESCRIPTION:
748 0810 1
749 0811 1     Output the compiled text corresponding to the format item
750 0812 1     just scanned
751 0813 1
752 0814 1 FORMAL PARAMETERS:
753 0815 1
754 0816 1     C      - format code
755 0817 1
756 0818 1 IMPLICIT INPUTS:
757 0819 1
758 0820 1     FMTDAT array
759 0821 1
760 0822 1 IMPLICIT OUTPUTS:
761 0823 1
762 0824 1     Compiled text output through argument
763 0825 1     Reinitialization for another format item (per format code related
764 0826 1     FMTDAT array updated)
765 0827 1
766 0828 1 ROUTINE VALUE:
767 0829 1
768 0830 1     NONE
769 0831 1
770 0832 1 SIDE EFFECTS:
771 0833 1
772 0834 1     SIGNAL_STOPs FOR$SYNERRFOR (62='SYNTAX ERROR IN FORMAT')
773 0835 1
774 0836 1 --
775 0837 1
776 0838 1
777 0839 2 BEGIN
778 0840 2 EXT_REG;                                ! Declare external registers
779 0841 2
780 0842 2 MACRO
781 M 0843 2 ALLBITS =
782 0844 2 0,0,32,0%,                                ! WHOLE WORD
783 M 0845 2 RSBITS =
784 0846 2 0,0,2,0%,                                ! REP COUNT SIZE
785 M 0847 2 SBIT =
786 0848 2 0,2,1,0%,                                ! W FIELD SIZE
787 M 0849 2 XBIT =
788 0850 2 0,7,1,0%;                                ! REPETITION COUNT EXISTS
789 0851 2
790 0852 2 MACRO
791 0853 2 ! Macro to pack flags for table FMT_PRM_LIMITS
792 0854 2
793 M 0855 2 FLAGBITS (F0, F1, F2, F3, F4, F5, F6, F7) =
794 M 0856 2
795 M 0857 2 (F0) OR (F1)^1 OR (F2)^2 OR (F3)^3 OR
796 0858 2 (F4)^4 OR (F5)^5 OR (F6)^6 OR (F7)^7 %
797 0859 2 ! Field definitions for table FMT_PRM_LIMITS
798 0860 2
799 M 0861 2 FDFLTOK =
800 0862 2 0,1,0%,                                ! Allows defaults if no parameters follow
```



```

801      M 0863 2      FMIN2 =
802      0864 2      1,1,0%,
803      M 0865 2      F1OR2 =
804      0866 2      2,1,0%,
805      M 0867 2      F2OR3 =
806      0868 2      3,1,0%,
807      M 0869 2      F1EXACT =
808      0870 2      4,1,0%,
809      0871 2      ! Macro to allow abbreviated reference to table FMT_PRM_LIMITS
810      0872 2
811      M 0873 2      FMT_CHECK (P0, S0, E0) =
812      0874 2      .FMT_PRM_LIMITS[C - TCODE, (P0), (S0), (E0)] %;
813      0875 2
814      0876 2      BIND
815      0877 2      Table of default options for parameters after a format edit
816      0878 2      specifier. Each row corresponds to an edit type.
817      0879 2      The bits are defined above. Edit specifiers not in the table
818      0880 2      (S, SS, SP, P, '(') do not allow following parameters.
819      0881 2
820      0882 2      FMT_PRM_LIMITS = UPLIT BYTE(
821      0883 2      FLAGBITS(0, 0, 0, 0, 1, 0, 0, 0),
822      0884 2      FLAGBITS(0, 0, 0, 0, 0, 0, 0, 0),
823      0885 2      FLAGBITS(0, 0, 0, 0, 0, 0, 0, 0),
824      0886 2      FLAGBITS(0, 0, 0, 0, 0, 0, 0, 0),
825      0887 2      FLAGBITS(0, 0, 0, 0, 0, 0, 0, 0),
826      0888 2      FLAGBITS(0, 0, 0, 0, 1, 0, 0, 0),
827      0889 2      FLAGBITS(0, 0, 0, 0, 1, 0, 0, 0),
828      0890 2      FLAGBITS(0, 0, 0, 0, 0, 0, 0, 0),
829      0891 2      FLAGBITS(1, 0, 0, 0, 0, 0, 0, 0),
830      0892 2      FLAGBITS(1, 0, 0, 0, 0, 0, 0, 0),
831      0893 2      FLAGBITS(1, 0, 1, 0, 0, 0, 0, 0),
832      0894 2      FLAGBITS(1, 0, 1, 0, 0, 0, 0, 0),
833      0895 2      FLAGBITS(1, 0, 1, 0, 0, 0, 0, 0),
834      0896 2      FLAGBITS(0, 0, 0, 0, 0, 0, 0, 0),
835      0897 2      FLAGBITS(0, 0, 0, 0, 0, 0, 0, 0),
836      0898 2      FLAGBITS(0, 0, 0, 0, 0, 0, 0, 0),
837      0899 2      FLAGBITS(0, 0, 0, 0, 0, 0, 0, 0),
838      0900 2      FLAGBITS(1, 1, 0, 0, 0, 0, 0, 0),
839      0901 2      FLAGBITS(1, 1, 0, 1, 0, 0, 0, 0),
840      0902 2      FLAGBITS(1, 1, 0, 1, 0, 0, 0, 0),
841      0903 2      FLAGBITS(1, 1, 0, 1, 0, 0, 0, 0),
842      0904 2      ) : BLOCK [, BYTE];
843      0905 2
844      0906 2      LOCAL
845      0907 2      FC : BLOCK [1],
846      0908 2      VFEM : BLOCK [1],
847      0909 2      VFEB;
848      0910 2
849      0911 2      ! If C is zero, there is nothing to reduce
850      0912 2      !
851      0913 2
852      0914 2      IF (FC = .C) NEQ 0
853      0915 2      THEN
854      0916 2      BEGIN
855      0917 2      ! Check whether this is a code which might have to be adjusted for
856      0918 2      ! a variable number of parameters
857      0919 2
```

```

858      0920      3
859      0921      3
860      0922      3
861      0923      4
862      0924      4
863      0925      4
864      0926      4
865      0927      4
866      0928      4
867      0929      4
868      0930      4
869      0931      4
870      0932      4
871      0933      4
872      0934      5
873      0935      5
874      0936      5
875      0937      5
876      0938      6
877      0939      6
878      0940      6
879      0941      6
880      0942      5
881      0943      5
882      0944      6
883      0945      6
884      0946      5
885      0947      4
886      0948      4
887      0949      4
888      0950      4
889      0951      4
890      0952      4
891      0953      4
892      0954      5
893      0955      5
894      0956      5
895      0957      5
896      0958      6
897      0959      6
898      0960      6
899      0961      6
900      0962      5
901      0963      4
902      0964      4
903      0965      4
904      0966      4
905      0967      4
906      0968      4
907      0969      4
908      0970      5
909      0971      5
910      0972      5
911      0973      5
912      0974      5
913      0975      4
914      0976      4

      IF .C GEQ TCODE
      THEN
      BEGIN
      ! The following block-IF checks for parameter consistency and calculates
      ! the correct format code for formats which allow variable numbers of
      ! parameters.
      !
      IF .SAVTYP [1] EQL 0
      THEN
      ! No parameters are present. If allowed, adjust format codes to
      ! indicate that defaults are being taken; otherwise, error.
      !
      BEGIN
      IF FMT_CHECK (FDFLTOK)
      THEN
      BEGIN
      FC = .FC + OFFSET;
      SAVTYP [2] = SAVTYP [3] = 0;
      END
      ELSE
      IF FMT_CHECK (F1EXACT) THEN ERROR (ERRFMTNUMB)
      END
      ELSE
      IF .SAVTYP [2] EQL 0
      THEN
      ! W field with no D field. This is an error for floating point
      ! edit types
      !
      BEGIN
      IF FMT_CHECK (FMIN2)
      THEN
      BEGIN
      ERROR (ERRFMTNUMB);
      END
      END
      ELSE
      IF .SAVTYP [3] EQL 0
      THEN
      ! W and D present, but not E. Check if this is W.M type and
      ! adjust format code if so.
      !
      BEGIN
      IF FMT_CHECK (F1OR2) THEN FC = .FC + IOZOFFSET
      END
      ELSE
      ! W,D,E present. If allowed, adjust format code, otherwise error.

```



```

915      0977      4      !
916      0978      4
917      0979      4      IF FMT_CHECK (F2OR3) THEN FC = .FC + EGOFFSET;
918      0980      4
919      0981      3      END;
920      0982      3
921      0983      3      IF .C EQL HCODE AND (.SAVVAL [1] LSS 0 OR .SAVTYP [1] LEQ 0) THEN ERROR (ERRHOLLCNT);
922      0984      3
923      0985      3      ! Compute the VFE-mask
924      0986      3
925      0987      3      VFEM [ALLBITS] = 0;
926      0988      3      ! Compute S and RS fields
927      0989      3
928      0990      3      ! If rep count is absent (SAVTYP[0] = 0), is a VFE, or is 1, then
929      0991      3      RSBIT = 0; otherwise it is the number of bytes necessary to
930      0992      3      represent the repetition count.
931      0993      3
932      0994      3
933      0995      3      IF .SAVTYP [0] LEQ 0 OR .SAVVAL [0] EQL 1
934      0996      3      THEN
935      0997      3          VFEM [RSBITS] = 0
936      0998      3      ELSE
937      0999      3          VFEM [RSBITS] = BYTSIZ (.SAVVAL [0]);
938      1000      3
939      1001      3      IF .C NEQ PCODE AND .SAVTYP [1] NEQ -1
940      1002      3      THEN
941      1003      3          VFEM [SBIT] = BYTSIZ (.SAVVAL [1]) - 1
942      1004      3      ELSE
943      1005      3          VFEM [SBIT] = 0;
944      1006      3
945      1007      3      VFEB = %0'200';
946      1008      3
947      1009      3      INCR I FROM 0 TO 3 DO
948      1010      4          BEGIN
949      1011      4              IF .SAVTYP [.I] LSS 0 THEN VFEM = .VFEM OR .VFEB;
950      1012      4
951      1013      4              VFEB = .VFEB^(-1);
952      1014      4
953      1015      4          END;
954      1016      3
955      1017      3      IF .VFEM [ALLBITS] NEQ 0 THEN FC [XBIT] = TRUE;
956      1018      3
957      1019      3      ! Output the code
958      1020      3      ! Also, check range of constant parameters
959      1021      3
960      1022      3      PUTBYT (.FC);
961      1023      3
962      1024      3      IF .VFEM [ALLBITS] NEQ 0 THEN PUTBYT (.VFEM [ALLBITS]);
963      1025      3
964      1026      3      INCR I FROM 0 TO 3 DO
965      1027      4
966      1028      4          CASE .SAVTYP [.I] FROM -1 TO 1 OF
967      1029      5              SET
968      1030      5              ! Case -1 Variable format expression
969      1031      5
970      1032      5
971      1033      5              [-1] :
```

```

: 972      1034      3
: 973      1035      3
: 974      1036      3
: 975      1037      3
: 976      1038      3
: 977      1039      3
: 978      1040      3
: 979      1041      3
: 980      1042      3
: 981      1043      3
: 982      1044      4
: 983      1045      4
: 984      1046      4
: 985      1047      4
: 986      1048      4
: 987      1049      4
: 988      1050      4
: 989      1051      4
: 990      1052      5
: 991      1053      5
: 992      1054      5
: 993      1055      5
: 994      1056      5
: 995      1057      5
: 996      1058      6
: 997      1059      6
: 998      1060      6
: 999      1061      6
1000      1062      6
1001      1063      5
1002      1064      5
1003      1065      4
1004      1066      4
1005      1067      4
1006      1068      4
1007      1069      4
1008      1070      5
1009      1071      5
1010      1072      5
1011      1073      5
1012      1074      5
1013      1075      5
1014      1076      5
1015      1077      6
1016      1078      5
1017      1079      5
1018      1080      5
1019      1081      5
1020      1082      5
1021      1083      5
1022      1084      5
1023      1085      5
1024      1086      5
1025      1087      5
1026      1088      5
1027      1089      4
: 1028      1090      4

      ERROR (ERRFMTCHAR);
      ! Case 0 Not present
      !
      [0] :
      ! 0;
      ! Case +1 Constant
      !
      [1] :
      BEGIN
      CASE .I FROM 0 TO 3 OF
      SET
      ! 0 - Repetition factor
      !
      [0] :
      BEGIN
      IF .SAVVAL [0] LEQ 0 THEN ERROR (ERRFMTRNGE);
      IF .SAVVAL [0] NEQ 1
      THEN
      BEGIN
      PUTBYT (.SAVVAL [0]);
      IF .VFEM [RSBITS] EQL 2 THEN PUTBYT (.SAVVAL [0]/256);
      END;
      END;
      ! 1 - Width or scaling factor
      !
      [1] :
      BEGIN
      IF .C EQL PCODE
      THEN
      IF .SAVVAL [1] LSS -128 OR .SAVVAL [1] GTR 127
      THEN
      ERROR (ERRFMTRNGE)
      ELSE
      0
      ELSE
      IF .SAVVAL [1] LSS 0 THEN ERROR (ERRFMTRNGE);
      PUTBYT (.SAVVAL [1]);
      IF .VFEM [SBIT] NEQ 0 THEN PUTBYT (.SAVVAL [1]/256);
      END;
      ! 2 - Decimal field width
```



```

: 1029      1091  4      !
: 1030      1092  4      !
: 1031      1093  4      [2] :
: 1032      1094  5      BEGIN
: 1033      1095  5      IF .SAVVAL [2] LSS 0 OR .SAVVAL [2] GTR 255 THEN ERROR (ERRFMTRNGE);
: 1034      1096  5      PUTBYT (.SAVVAL [2]);
: 1035      1097  5      END;
: 1036      1098  5      ! 3 - Exponent field
: 1037      1099  4      !
: 1038      1100  4      !
: 1039      1101  4      [3] :
: 1040      1102  4      BEGIN
: 1041      1103  5      IF .SAVVAL [3] LSS 0 OR .SAVVAL [3] GTR 255 THEN ERROR (ERRFMTRNGE);
: 1042      1104  5      PUTBYT (.SAVVAL [3]);
: 1043      1105  5      END;
: 1044      1106  5      TES;
: 1045      1107  5      END;
: 1046      1108  5      TES;
: 1047      1109  4      END;
: 1048      1110  4      TES;
: 1049      1111  4      END;
: 1050      1112  4      TES;
: 1051      1113  3      END;
: 1052      1114  3      TES;
: 1053      1115  2      END;
: 1054      1116  2      CH$FILL (0, %UPVAL*(K_PTR_OFFSET + L_NEST), SAVVAL [0]); ! Zero to but not including NEST
: 1055      1117  2      END;
: 1056      1118  1
```

```

00 00 05 05 05 01 01 00 10 10 00 00 00 00 10 00418 P.AAB: .BYTE 16, 0, 0, 0, 0, 16, 16, 0, 1, 1, 5, 5, 5, - ;
OB OB OB 03 00 00 00427 0, 0, 0, 0, 3, 11, 11, 11 ;
```

FMT_PRM_LIMITS= P.AAB

```

56      0000V 007C 00000 REDUCE: .WORD Save R2,R3,R4,R5,R6 : 0806
55      04 AC D0 00002 MOVAB PUTBYT, R6 :
53      55 D0 0000B MOVL C, R5 : 0914
      03 12 0000E MOVL R5, FC
      0168 31 00010 BNEQ 1$
0D      55 D1 00013 1$: BRW 34$
      39 19 00016 CMPL R5, #13 : 0921
52      C2 AF45 9E 00018 BLSS 8$
      04 AA D5 0001D MOVAB FMT_PRM_LIMITS-13[R5], R2 : 0936
      11 12 00020 TSTL 4(SAVTYP) : 0929
08      62 E9 00022 BNEQ 3$
53      14 C0 00025 BLBC (R2), 2$ : 0936
      08 AA 7C 00028 ADDL2 #20, FC : 0939
      24 11 0002B CLRQ 8(SAVTYP) : 0940
      04 E1 0002D BRB 8$ : 0934
20      62 09 11 00031 2$: BBC #4, (R2), 8$ : 0944
      08 AA D5 00033 3$: TSTL 8(SAVTYP) : 0949
      07 12 00036 BNEQ 5$
15      62 01 E1 00038 BBC #1, (R2), 8$ : 0956
```

			0127	31	0003C	4\$:	BRW	30\$:	0959	
			0C	AA	D5	0003F	5\$:	TSTL	12(SAVTYP)	:	0965
				06	12	00042		BNEQ	6\$:	
	09	62		02	E1	00044		BBC	#2, (R2), 8\$:	0972
				04	11	00048		BRB	7\$:	
	03	62		03	E1	0004A	6\$:	BBC	#3, (R2), 8\$:	0979
		53		03	C0	0004E	7\$:	ADDL2	#3, FC	:	
		0F		55	D1	00051	8\$:	CMPL	R5, #15	:	0983
				0A	12	00054		BNEQ	9\$:	
			04	AB	D5	00056		TSTL	4(SAVVAL)	:	
				E1	19	00059		BLSS	4\$:	
			04	AA	D5	0005B		TSTL	4(SAVTYP)	:	
				DC	15	0005E		BLEQ	4\$:	
				54	D4	00060	9\$:	CLRL	VFEM	:	0987
				6A	D5	00062		TSTL	(SAVTYP)	:	0995
				05	15	00064		BLEQ	10\$:	
		01		6B	D1	00066		CMPL	(SAVVAL), #1	:	
				05	12	00069		BNEQ	11\$:	
		54		03	8A	0006B	10\$:	BICB2	#3, VFEM	:	0997
				0C	11	0006E		BRB	12\$:	
				6B	DD	00070	11\$:	PUSHL	(SAVVAL)	:	0999
				01	FB	00072		CALLS	#1, BYTSIZ	:	
54	02	0000V	CF	50	F0	00077		INSV	R0, #0, #2, VFEM	:	
			0C	55	D1	0007C	12\$:	CMPL	R5, #12	:	1001
				1D	13	0007F		BEQL	13\$:	
		FFFFFFF	8F	04	AA	D1	00081	CMPL	4(SAVTYP), #-1	:	
				13	13	00089		BEQL	13\$:	
				04	AB	DD	0008B	PUSHL	4(SAVVAL)	:	1003
		0000V	CF	01	FB	0008E		CALLS	#1, BYTSIZ	:	
54	01		51	FF	A0	9E	00093	MOVAB	-1(R0), R1	:	
			02		51	F0	00097	INSV	R1, #2, #1, VFEM	:	
					03	11	0009C	BRB	14\$:	
		54			04	8A	0009E	13\$:	BICB2	#4, VFEM	1005
		51		80	8F	9A	000A1	14\$:	MOVZBL	#128, VFEB	1007
					50	D4	000A5		CLRL	I	1009
				6A40	D5	000A7	15\$:	TSTL	(SAVTYP)[I]	:	1012
					03	18	000AA	BGEQ	16\$:	
		54			51	C8	000AC	BISL2	VFEB, VFEM	:	
51		51		FF	8F	78	000AF	16\$:	ASHL	#-1, VFEB, VFEB	1014
EF		50			03	F3	000B4	A0BLEQ	#3, I, 15\$:	1009
					52	D4	000B8		CLRL	R2	1017
					54	D5	000BA		TSTL	VFEM	
					06	13	000BC		BEQL	17\$	
					52	D6	000BE		INCL	R2	
		53		80	8F	88	000C0		BISB2	#128, FC	
					53	DD	000C4	17\$:	PUSHL	FC	1022
		66			01	FB	000C6		CALLS	#1, PUTBYT	
		05			52	E9	000C9		BLBC	R2, 18\$	1024
					54	DD	000CC		PUSHL	VFEM	
		66			01	FB	000CE		CALLS	#1, PUTBYT	
					53	D4	000D1	18\$:	CLRL	I	1026
02	FFFFFFF	8F		6A43	CF	000D3	19\$:	CASEL	(SAVTYP)[I], #-1, #2	:	1028
0008		0099		008A		000DC	20\$:	.WORD	30\$-20\$,-	:	
									33\$-20\$,-	:	
									21\$-20\$:	
									25\$:	1034
03		00		46	11	000E2		BRB		:	
				53	CF	000E4	21\$:	CASEL	I, #0, #3	:	1046

006F	005E	0027	0008	000E8 22\$:	.WORD	23\$-22\$,-	
			6B D5 000F0	23\$:	TSTL	(SAVVAL)	1054
			72 15 000F2		BLEQ	30\$	
		01	6B D1 000F4		CMPL	(SAVVAL), #1	1056
			7C 13 000F7		BEQL	33\$	
			6B DD 000F9		PUSHL	(SAVVAL)	1059
02		66	01 FB 000FB		CALLS	#1, PUTBYT	
	54	02	00 ED 000FE		CMPZV	#0, #2, VFEM, #2	1061
			70 12 00103		BNEQ	33\$	
	7E	6B 00000100	8F C7 00105		DIVL3	#256, (SAVVAL), -(SP)	
			63 11 0010D		BRB	32\$	
		0C	55 D1 0010F	24\$:	CMPL	R5, #12	1072
			18 12 00112		BNEQ	26\$	
		52 04	AB D0 00114		MOVL	4(SAVVAL), R2	1075
	FFFFFFF80	8F	52 D1 00118		CMPL	R2, #-128	
			45 19 0011F		BLSS	30\$	
	0000007F	8F	52 D1 00121		CMPL	R2, #127	
			08 15 00128		BLEQ	27\$	
		52 04	3A 11 0012A	25\$:	BRB	30\$	1077
			AB D0 0012C	26\$:	MOVL	4(SAVVAL), R2	1083
			34 19 00130		BLSS	30\$	
			52 DD 00132	27\$:	PUSHL	R2	1085
		66	01 FB 00134		CALLS	#1, PUTBYT	
	3A	54	02 E1 00137		BBC	#2, VFEM, 33\$	1087
	7E 04	AB 00000100	8F C7 0013B		DIVL3	#256, 4(SAVVAL), -(SP)	
			2C 11 00144		BRB	32\$	
		52 08	AB D0 00146	28\$:	MOVL	8(SAVVAL), R2	1096
			1A 19 0014A		BLSS	30\$	
	000000FF	8F	52 D1 0014C		CMPL	R2, #255	
			11 14 00153		BGTR	30\$	
		52 0C	19 11 00155		BRB	31\$	1098
			AB D0 00157	29\$:	MOVL	12(SAVVAL), R2	1106
			09 19 0015B		BLSS	30\$	
	000000FF	8F	52 D1 0015D		CMPL	R2, #255	
			0A 15 00164		BLEQ	31\$	
		00000000G	3E DD 00166	30\$:	PUSHL	#62	
			01 FB 00168		CALLS	#1, FOR\$\$SIGNAL_STO	
			04 0016F		RET		
		66	52 DD 00170	31\$:	PUSHL	R2	1108
		01	01 FB 00172	32\$:	CALLS	#1, PUTBYT	
FF58	53	01	03 F1 00175	33\$:	ACBL	#3, #1, 1, 19\$	1028
2C	00	6E	00 2C 0017B	34\$:	MOVC5	#0, (SP), #0, #44, (SAVVAL)	1117
			6B 00180				
			04 00181		RET		1118

; Routine Size: 386 bytes, Routine Base: _FOR\$CODE + 042D

```
1058 1119 1 ROUTINE DEFER (C, N) : CALL_G3 NOVALUE =
1059 1120 1
1060 1121 1 ++
1061 1122 1 FUNCTIONAL DESCRIPTION:
1062 1123 1
1063 1124 1 Cuase the reduction of the current format item to be deferred
1064 1125 1 until the W.D portion has been read
1065 1126 1
1066 1127 1 FORMAL PARAMETERS:
1067 1128 1
1068 1129 1 C - format code
1069 1130 1 N - number of parameters to follow (1 or 2)
1070 1131 1
1071 1132 1 IMPLICIT INPUTS:
1072 1133 1
1073 1134 1 FMTDAT array
1074 1135 1
1075 1136 1
1076 1137 1 IMPLICIT OUTPUTS:
1077 1138 1
1078 1139 1 repetition count, if any, saved in FMTDAT
1079 1140 1 format code and parameter count saved also in FMTDAT
1080 1141 1
1081 1142 1 ROUTINE VALUE:
1082 1143 1
1083 1144 1 NONE
1084 1145 1
1085 1146 1 SIDE EFFECTS:
1086 1147 1
1087 1148 1 SIGNAL_STOPs FOR$SYNERRFOR (62='SYNTAX ERROR IN FORMAT')
1088 1149 1
1089 1150 1 --
1090 1151 1
1091 1152 2 BEGIN
1092 1153 2 EXT REG; ! Declare external registers
1093 1154 2 NZERO ();
1094 1155 2 NSAVE ();
1095 1156 2 PTR [L_FDEFER] = .C;
1096 1157 2 PTR [L_FCOUNT] = .N;
1097 1158 1 END;
```

0000V	CF	00	FB	00002	DEFER:	.WORD	Save nothing	: 1119
0000V	CF	00	FB	00007		CALLS	#0, NZERO	: 1154
	69	04	AC	7D 0000C		CALLS	#0, NSAVE	: 1155
				04 00010		MOVQ	C, (PTR)	: 1156
						RET		: 1158

; Routine Size: 17 bytes, Routine Base: _FOR\$CODE + 05AF


```
1099 1159 1 ROUTINE UNDEFER : CALL_G3 NOVALUE =
1100 1160 1
1101 1161 1 ++
1102 1162 1 FUNCTIONAL DESCRIPTION:
1103 1163 1
1104 1164 1 Complete the reduction of a format item which was deferred
1105 1165 1
1106 1166 1 FORMAL PARAMETERS:
1107 1167 1
1108 1168 1
1109 1169 1 IMPLICIT INPUTS:
1110 1170 1
1111 1171 1 FMTDAT array
1112 1172 1
1113 1173 1 IMPLICIT OUTPUTS:
1114 1174 1
1115 1175 1 FMTDAT array
1116 1176 1
1117 1177 1 ROUTINE VALUE:
1118 1178 1
1119 1179 1 NONE
1120 1180 1
1121 1181 1 SIDE EFFECTS:
1122 1182 1
1123 1183 1 SIGNAL_STOPs FOR$SYNERRFOR (62='SYNTAX ERROR IN FORMAT')
1124 1184 1
1125 1185 1 --
1126 1186 1
1127 1187 2 BEGIN
1128 1188 2 EXT_REG; ! Declare external registers
1129 1189 2
1130 1190 2 IF .PTR [L_FDEFER] NEQ 0
1131 1191 2 THEN
1132 1192 3 BEGIN
1133 1193 3 NSAVE ();
1134 1194 3 REDUCE (.PTR [L_FDEFER]);
1135 1195 3 END
1136 1196 2 ELSE
1137 1197 3 BEGIN
1138 1198 3
1139 1199 3 IF .PTR [L_TYPE] NEQ 0 THEN ERROR (ERRFMXTNUM);
1140 1200 3
1141 1201 3 IF .PTR [L_SIGN] NEQ 0 THEN ERROR (ERRFMTCHAR);
1142 1202 3
1143 1203 3 PTR [L_NVAL] = 0;
1144 1204 3 PTR [L_TYPE] = 0;
1145 1205 3 PTR [L_SIGN] = 0;
1146 1206 2 END;
1147 1207 2
1148 1208 1 END;
```

```
0000 00000 UNDEFER: .WORD Save nothing
69 D5 00002 YSTL (PTR)
```

```
: 1159
: 1190
```

0000V	CF	0D	13	00004	BEQL	1\$:	1193
		00	FB	00006	CALLS	#0, NSAVE	:	1194
FE5B	CF	69	DD	0000B	PUSHL	(PTR)	:	1190
		01	FB	0000D	CALLS	#1, REDUCE	:	1199
			04	00012	RET		:	1201
		18	A9	D5 00013	TSTL	24(PTR)	:	
			05	12 00016	BNEQ	2\$:	
		10	A9	D5 00018	TSTL	16(PTR)	:	
			0A	13 0001B	BEQL	3\$:	
00000000G	00	3E	DD	0001D	PUSHL	#62	:	
		01	FB	0001F	CALLS	#1, FOR\$\$SIGNAL_STO	:	
			04	00026	RET		:	
		14	A9	7C 00027	CLRQ	20(PTR)	:	1203
		10	A9	D4 0002A	CLRL	16(PTR)	:	1205
			04	0002D	RET		:	1208

; Routine Size: 46 bytes, Routine Base: _FOR\$CODE + 05C0


```
: 1150      1209 1 ROUTINE NZERO : CALL_G3 NOVALUE =
: 1151      1210 1
: 1152      1211 1
: 1153      1212 1 ++
: 1154      1213 1 FUNCTIONAL DESCRIPTION:
: 1155      1214 1      Check context for a format item with has an optional leading
: 1156      1215 1      number field. If there is a deferred item, then a separator is
: 1157      1216 1      required, and we have an ambiguous case. The leading numeric
: 1158      1217 1      will be attached to the preceding format item.
: 1159      1218 1
: 1160      1219 1 FORMAL PARAMETERS:
: 1161      1220 1
: 1162      1221 1      None
: 1163      1222 1
: 1164      1223 1 IMPLICIT INPUTS:
: 1165      1224 1
: 1166      1225 1      FMTDAT array
: 1167      1226 1
: 1168      1227 1
: 1169      1228 1 IMPLICIT OUTPUTS:
: 1170      1229 1
: 1171      1230 1      NONE
: 1172      1231 1
: 1173      1232 1 ROUTINE VALUE:
: 1174      1233 1
: 1175      1234 1      NONE
: 1176      1235 1
: 1177      1236 1 SIDE EFFECTS:
: 1178      1237 1
: 1179      1238 1      SIGNAL_STOPs FOR$$SYNERRFOR (62="SYNTAX ERROR IN FORMAT")
: 1180      1239 1
: 1181      1240 1 --
: 1182      1241 1
: 1183      1242 2 BEGIN
: 1184      1243 2 EXT_REG;                      ! Declare external registers
: 1185      1244 2
: 1186      1245 2 IF .PTR [L_FDEFER] NEQ 0 THEN ERROR (ERRFMTSEPR);
: 1187      1246 2
: 1188      1247 1 END;
```

	0000 00000	NZERO:	.WORD	Save nothing	: 1209
69	D5 00002		TSTL	(PTR)	: 1245
09	13 00004		BEQL	1\$:
3E	DD 00006		PUSHL	#62	:
01	FB 00008		CALLS	#1, FOR\$\$SIGNAL_STO	:
	04 0000F	1\$:	RET		: 1247

; Routine Size: 16 bytes, Routine Base: _FOR\$CODE + 05EE

```
: 1190      1248 1 ROUTINE NSAVE : CALL_G3 NOVALUE =
: 1191      1249 1
: 1192      1250 1
: 1193      1251 1 ++
: 1194      1252 1 FUNCTIONAL DESCRIPTION:
: 1195      1253 1     Save the values of PTR[L_NVAL] and PTR[L_TYPE] in SAVTYP and SAVVAL
: 1196      1254 1
: 1197      1255 1 FORMAL PARAMETERS:
: 1198      1256 1
: 1199      1257 1     None
: 1200      1258 1
: 1201      1259 1 IMPLICIT INPUTS:
: 1202      1260 1
: 1203      1261 1     PTR[L_NVAL]      - value of a numeric term
: 1204      1262 1     PTR[L_TYPE]     - PTR[L_TYPE] of the numeric term
: 1205      1263 1     PTR[L_SIGN]    - indicator if a minus PTR[L_SIGN] has been encountered
: 1206      1264 1     PTR[L_PHASE]   - indicator of what the PTR[L_NVAL] and PTR[L_TYPE] associate
: 1207      1265 1     to repetition count, W or D.
: 1208      1266 1
: 1209      1267 1 IMPLICIT OUTPUTS:
: 1210      1268 1
: 1211      1269 1     FMTDAT array
: 1212      1270 1
: 1213      1271 1 ROUTINE VALUE:
: 1214      1272 1
: 1215      1273 1     NONE
: 1216      1274 1
: 1217      1275 1 SIDE EFFECTS:
: 1218      1276 1
: 1219      1277 1     SIGNAL_STOPs FOR$SYNERRFOR (62='SYNTAX ERROR IN FORMAT')
: 1220      1278 1
: 1221      1279 1 --
: 1222      1280 1
: 1223      1281 2 BEGIN
: 1224      1282 2 EXT_REG;                                ! Declare external registers
: 1225      1283 2
: 1226      1284 2 IF .PTR [L_SIGN] NEQ 0 THEN ERROR (ERRFMTPTR [L_SIGN]);
: 1227      1285 2
: 1228      1286 2 SAVVAL [.PTR [L_PHASE]] = .PTR [L_NVAL];
: 1229      1287 2 SAVTYP [.PTR [L_PHASE]] = .PTR [L_TYPE];
: 1230      1288 2 PTR [L_PHASE] = .PTR [L_PHASE] + 1;
: 1231      1289 2 PTR [L_SIGN] = 0;
: 1232      1290 2 PTR [L_NVAL] = 0;
: 1233      1291 2 PTR [L_TYPE] = 0;
: 1234      1292 1 END;
```

			0000 00000 NSAVE:	.WORD	Save nothing	: 1248
		10	A9 D5 00002	ISTL	16(PTR)	: 1284
			0A 13 00005	BEQL	1\$:
			3E DD 00007	PUSHL	#62	:
00000000G	00		01 FB 00009	CALLS	#1, FOR\$SIGNAL_STO	:
			04 00010	RET		:
	50	08	A9 D0 00011 1\$:	MOVL	8(PTR), R0	: 1286

FOR\$\$FMTCP
2-006

FORTTRAN OBJECT TIME FORMAT COMPILER

I 12
16-Sep-1984 00:23:29
14-Sep-1984 12:31:59

VAX-11 Bliss-32 V4.0-742
[FORRTL.SRC]FORFMTCP.B32;1

Page 35
(8)

6B40	14	A9	D0	00015
6A40	18	A9	D0	0001A
	08	A9	D6	0001F
	10	A9	7C	00022
	18	A9	D4	00025
			04	00028

MOVL	20(PTR),	(SAVVAL)[R0]
MOVL	24(PTR),	(SAVTYP)[R0]
INCL	8(PTR)	
CLRQ	16(PTR)	
CLRL	24(PTR)	
RET		

:	1287
:	1288
:	1289
:	1291
:	1292

; Routine Size: 41 bytes, Routine Base: _FOR\$CODE + 05FE

FOR
2-0

```
: 1236 1293 1 ROUTINE PUTBYT (V) : CALL_G3 NOVALUE =
: 1237 1294 1
: 1238 1295 1 !++
: 1239 1296 1 FUNCTIONAL DESCRIPTION:
: 1240 1297 1
: 1241 1298 1     Output a byte through argument
: 1242 1299 1
: 1243 1300 1 FORMAL PARAMETERS:
: 1244 1301 1
: 1245 1302 1     V     - vaule to be output
: 1246 1303 1
: 1247 1304 1 IMPLICIT INPUTS:
: 1248 1305 1
: 1249 1306 1     FMTDAT array
: 1250 1307 1
: 1251 1308 1 IMPLICIT OUTPUTS:
: 1252 1309 1
: 1253 1310 1     FMTDAT array
: 1254 1311 1
: 1255 1312 1 ROUTINE VALUE:
: 1256 1313 1
: 1257 1314 1     NONE
: 1258 1315 1
: 1259 1316 1 SIDE EFFECTS:
: 1260 1317 1
: 1261 1318 1     SIGNAL_STOPs FOR$SYNERRFOR (62='SYNTAX ERROR IN FORMAT')
: 1262 1319 1
: 1263 1320 1 !--
: 1264 1321 1
: 1265 1322 2 BEGIN
: 1266 1323 2
: 1267 1324 2 LOCAL
: 1268 1325 2     A_OLD_BUF_BEG;           ! Place to save old format buffer address
: 1269 1326 2
: 1270 1327 2 EXT_REG;             ! Declare external registers
: 1271 1328 2
: 1272 1329 2 !+
: 1273 1330 2 ! Check if room in currently allocated format buffer.
: 1274 1331 2 ! If not allocate twice as much and copy old format buffer,
: 1275 1332 2 ! then deallocate old format buffer.
: 1276 1333 2 !-
: 1277 1334 2
: 1278 1335 2 IF .PTR [L_NCHAR] GEQ .PTR [L_FMT_BUF_SIZ]
: 1279 1336 2 THEN
: 1280 1337 3 BEGIN
: 1281 1338 3     A_OLD_BUF_BEG = .PTR [A_FMT_BUF_BEG];
: 1282 1339 3
: 1283 1340 3     IF .PTR [L_FMT_BUF_SIZ] GEQ 32768 THEN ERROR ();
: 1284 1341 3
: 1285 1342 3     PTR [A_FMT_BUF_BEG] = FOR$$GET_VM (.PTR [L_FMT_BUF_SIZ]*2);
: 1286 1343 3     CH$MOVE (.PTR [L_FMT_BUF_SIZ], .A_OLD_BUF_BEG, .PTR [A_FMT_BUF_BEG]);
: 1287 1344 3     FOR$$FREE_VM (.PTR [L_FMT_BUF_SIZ], .A_OLD_BUF_BEG);
: 1288 1345 3     PTR [L_FMT_BUF_SIZ] = .PTR [L_FMT_BUF_SIZ]*2;
: 1289 1346 3 END;
: 1290 1347 2
: 1291 1348 2 !+
: 1292 1349 2 ! Store away the byte in format buffer
```



```
: 1293      1350  2      !-  
: 1294      1351  2  
: 1295      1352  2      (.PTR [A_FMT_BUF_BEG] + .PTR [L_NCHAR]) < 0, 8 > = .V;  
: 1296      1353  2      PTR [L_NCHAR] = .PTR [L_NCHAR] + 1;  
: 1297      1354  1      END;
```

				007C 00000	PUTBYT: .WORD	Save R2,R3,R4,R5,R6	: 1293
	28	A9	1C	A9 D1 00002	CMPL	28(PTR), 40(PTR)	: 1335
				3E 19 00007	BLSS	2\$	
		56	20	A9 D0 00009	MOVL	32(PTR), A_OLD_BUF_BEG	: 1338
	00008000	8F	28	A9 D1 0000D	CMPL	40(PTR), #32768	: 1340
				0A 19 00015	BLSS	1\$	
				3E DD 00017	PUSHL	#62	
	00000000G	00		01 FB 00019	CALLS	#1, FOR\$\$SIGNAL_STO	
				04 00020	RET		
	7E	28		01 78 00021	ASHL	#1, 40(PTR), -(SP)	: 1342
	00000000G	00		01 FB 00026	CALLS	#1, FOR\$\$GET_VM	
		20		50 D0 0002D	MOVL	R0, 32(PTR)	
20	B9			A9 28 00031	MOVC3	40(PTR), (A_OLD_BUF_BEG), @32(PTR)	: 1343
		66	28	56 DD 00037	PUSHL	A_OLD_BUF_BEG	: 1344
			28	A9 DD 00039	PUSHL	40(PTR)	
	00000000G	00		02 FB 0003C	CALLS	#2, FOR\$\$FREE_VM	
		28		02 C4 00043	MULL2	#2, 40(PTR)	: 1345
50		20	1C	A9 C1 00047	ADDL3	28(PTR), 32(PTR), R0	: 1352
		60	04	AC 90 0004D	MOVB	V, (R0)	
			1C	A9 D6 00051	INCL	28(PTR)	: 1353
				04 00054	RET		: 1354

; Routine Size: 85 bytes, Routine Base: _FOR\$CODE + 0627

```
: 1299 1355 1 ROUTINE BYTSIZ (VAL) =
: 1300 1356 1
: 1301 1357 1 ++
: 1302 1358 1 FUNCTIONAL DESCRIPTION:
: 1303 1359 1
: 1304 1360 1 Calculate the number of bytes to hold VAL
: 1305 1361 1
: 1306 1362 1 FORMAL PARAMETERS:
: 1307 1363 1
: 1308 1364 1 VAL - value to be sized
: 1309 1365 1
: 1310 1366 1 IMPLICIT INPUTS:
: 1311 1367 1
: 1312 1368 1 NONE
: 1313 1369 1
: 1314 1370 1
: 1315 1371 1 IMPLICIT OUTPUTS:
: 1316 1372 1
: 1317 1373 1 NONE
: 1318 1374 1
: 1319 1375 1 ROUTINE VALUE:
: 1320 1376 1
: 1321 1377 1 NONE
: 1322 1378 1
: 1323 1379 1 SIDE EFFECTS:
: 1324 1380 1
: 1325 1381 1 SIGNAL_STOPs FOR$SYNERRFOR (62='SYNTAX ERROR IN FORMAT')
: 1326 1382 1
: 1327 1383 1 --
: 1328 1384 1
: 1329 1385 2 BEGIN
: 1330 1386 2
: 1331 1387 2 MAP
: 1332 1388 2 VAL : LONG UNSIGNED;
: 1333 1389 2
: 1334 1390 2 IF .VAL LSS 0
: 1335 1391 2 THEN
: 1336 1392 3 ERROR (ERRFMTRNGE)
: 1337 1393 2 ELSE
: 1338 1394 2
: 1339 1395 2 IF .VAL LSS 256
: 1340 1396 2 THEN
: 1341 1397 2 RETURN 1
: 1342 1398 2 ELSE
: 1343 1399 2
: 1344 1400 2 IF .VAL LSS 65536 THEN RETURN 2 ELSE ERROR (ERRFMTRNGE);
: 1345 1401 2
: 1346 1402 1 END;
```

```
00000100 52 04 0004 0000 BYTSIZ: .WORD Save R2
AC D0 00002 MOVL VAL, R2
1A 19 00006 BLSS 2$
52 D1 00008 CMPL R2, #256
```

```
: 1355
: 1390
: 1395
```


	50	04 18 0000F	BGEQ 1\$
		01 D0 00011	MOVL #1, R0
		04 04 00014	RET
00010000	8F	52 D1 00015 1\$:	CMPL R2, #65536
		04 18 0001C	BGEQ 2\$
	50	02 D0 0001E	MOVL #2, R0
		04 04 00021	RET
00000000G	00	3E DD 00022 2\$:	PUSHL #62
		01 FB 00024	CALLS #1, FOR\$\$SIGNAL_STO
		50 D4 0002B	CLRL R0
		04 04 0002D	RET

: 1397
: 1400
: 1402

: Routine Size: 46 bytes, Routine Base: _FOR\$CODE + 067C

: 1347 1403 1 END
: 1348 1404 1
: 1349 1405 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
_FOR\$CODE	1706	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	0	0	581	00:01.0
_\$255\$DUA28:[FORRTL.OBJ]FORLIB.L32;1	711	2	0	52	00:00.5
_\$255\$DUA28:[FORRTL.OBJ]RTLLIB.L32;1	36	0	0	8	00:00.1

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD, INITIAL, OPTIMIZE)/NOTRACE/LIS=LISS\$:FORFMTCP/OBJ=OBJ\$:FORFMTCP MSRC\$:FORFMTCP/UPDATE=(ENH\$:FORFMTCP)

: Size: 1594 code + 112 data bytes
: Run Time: 00:36.2
: Elapsed Time: 01:28.9
: Lines/CPU Min: 2326
: Lexemes/CPU-Min: 16995

FOR\$\$FMTCP
2-006

FORTTRAN OBJECT TIME FORMAT COMPILER

N 12
16-Sep-1984 00:23:29

VAX-11 Bliss-32 V4.0-742

Page 40

: Memory Used: 326 pages
: Compilation Complete

FOR
2-(

0180 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

